日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　６月１５日

出願番号
Application Number: 特願２００１－１８１７９１

出願人
Applicant(s): 株式会社東芝

CERTIFIED COPY OF
PRIORITY DOCUMENT

２００１年１２月２８日

特許庁長官
Commissioner,
Japan Patent Office 及川耕造

出証番号　出証特２００１－３１１２８３５

【書類名】　特許願

【整理番号】　A000102769

【提出日】　平成13年 6月15日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 9/00

【発明の名称】　分散システムおよび同システムの多重化制御方法

【請求項の数】　8

【発明者】

【住所又は居所】　東京都府中市東芝町１番地　株式会社東芝府中事業所内

【氏名】　遠藤　浩太郎

【特許出願人】

【識別番号】　000003078

【氏名又は名称】　株式会社　東芝

【代理人】

【識別番号】　100058479

【弁理士】

【氏名又は名称】　鈴江　武彦

【電話番号】　03-3502-3181

【選任した代理人】

【識別番号】　100084618

【弁理士】

【氏名又は名称】　村松　貞男

【選任した代理人】

【識別番号】　100068814

【弁理士】

【氏名又は名称】　坪井　淳

【選任した代理人】

【識別番号】　100092196

【弁理士】

【氏名又は名称】　橋本　良郎

【選任した代理人】

【識別番号】　100091351

【弁理士】

【氏名又は名称】　河野　哲

【選任した代理人】

【識別番号】　100088683

【弁理士】

【氏名又は名称】　中村　誠

【選任した代理人】

【識別番号】　100070437

【弁理士】

【氏名又は名称】　河井　将次

【手数料の表示】

【予納台帳番号】　011567

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　　分散システムおよび同システムの多重化制御方法

【特許請求の範囲】

【請求項１】　ネットワークで接続されたｎ台のコンピュータを同期的に動作させる分散システムであって、少なくとも（ｎ－ｆ）台以上での多重化を保証する分散システムにおいて、

前記コンピュータそれぞれは、

各々が次に処理する候補として選択した入力データを前記ネットワークを介して収集する入力候補収集手段と、

前記入力候補収集手段により収集された入力データが（ｎ－ｆ）個以上存在する場合に、その中に同一内容の入力データが（ｎ－ｆ）個以上あるか否かを判定し、（ｎ－ｆ）個以上あったときに、その入力データを次に処理する対象として確定する第１の入力候補選定制御手段と、

前記収集された入力データの中に同一内容の入力データが（ｎ－ｆ）個以上なかったときに、前記収集された入力データ数の過半数を占める同一内容の入力データが存在するか否かを判定し、存在したときに、その入力データを自候補とするとともにそれ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第２の入力候補選定制御手段と、

前記収集された入力データ数の過半数を占める同一内容の入力データが存在しなかったときに、前記収集された入力データの中からいずれかの入力データを任意に選択して自候補とするとともに、それ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第３の入力候補選定制御手段と

を具備することを特徴とする分散システム。

【請求項２】　ｆは、３ｆ＜ｎを満たす最大の整数であることを特徴とする請求項１記載の分散システム。

【請求項３】　前記コンピュータそれぞれは、

前記第１の入力候補確定制御手段により確定された入力データを保持するジャーナル手段と、

自コンピュータで既に確定済みである工程の入力データを他のコンピュータが収集しているときに、前記ジャーナル手段に保持された入力データを確定済みの入力データとして送信する第１の入力候補調整制御手段と、

前記入力候補収集手段による入力データの収集時に、他のコンピュータから確定済みの入力データが送信されたときに、その入力データを次に処理する対象として確定する第２の入力候補調整制御手段と

をさらに具備することを特徴とする請求項１または２記載の分散システム。

【請求項４】　前記ジャーナル手段は、前記入力データを最新のものから予め定められた工程数だけ保持し、

前記第１の入力候補調整制御手段は、他のコンピュータに送信すべき確定済みの入力データが前記ジャーナル手段に保持されていないときに、その旨を通知する手段を具備し、

前記コンピュータそれぞれは、

自コンピュータで既に確定済みである各工程での直前の状態を予め定められた工程分まで保持する状態保持手段と、

前記状態保持手段に保持された各工程での直前の状態を他のコンピュータとの間で送受信する状態送受信手段と、

前記入力候補収集手段による入力データの収集時に、その収集された入力データ数と他のコンピュータから確定済みの入力データが前記ジャーナル手段にも既に保持されていない旨を通知された数との和が（ｎ－ｆ）以上であって、前記収集された入力データ数が（ｎ－ｆ）未満であったときに、他のすべてのコンピュータの中で確定済みの工程が最も進んだ他のコンピュータにおける最新の確定済みの工程での直前の状態を前記状態送受信手段により取得して自コンピュータに複製するスキップ手段と

をさらに具備することを特徴とする請求項３記載の分散システム。

【請求項５】　ネットワークで接続されたｎ台のコンピュータを同期的に動作させる分散システムであって、少なくとも（ｎ－ｆ）台以上での多重化を保証する分散システムの多重化制御方法であって、

各々が次に処理する候補として選択した入力データを前記ネットワークを介し

て収集する入力候補収集ステップと、

前記入力候補収集ステップにより収集された入力データが（ｎ－ｆ）個以上存在する場合に、その中に同一内容の入力データが（ｎ－ｆ）個以上あるか否かを判定し、（ｎ－ｆ）個以上あったときに、その入力データを次に処理する対象として確定する第１の入力候補選定制御ステップと、

前記収集された入力データの中に同一内容の入力データが（ｎ－ｆ）個以上なかったときに、前記収集された入力データ数の過半数を占める同一内容の入力データが存在するか否かを判定し、存在したときに、その入力データを自候補とするとともにそれ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集ステップに入力データの収集を再実行させる第２の入力候補選定制御ステップと、

前記収集された入力データ数の過半数を占める同一内容の入力データが存在しなかったときに、前記収集された入力データの中からいずれかの入力データを任意に選択して自候補とするとともに、それ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第３の入力候補選定制御ステップと

を具備することを特徴とする分散システムの多重化制御方法。

【請求項６】　ｆは、３ｆ＜ｎを満たす最大の整数であることを特徴とする請求項５記載の分散システムの多重化制御方法。

【請求項７】　前記第１の入力候補確定制御ステップにより確定された入力データを保持するジャーナルステップと、

自コンピュータで既に確定済みである工程の入力データを他のコンピュータが収集しているときに、前記ジャーナルステップにより保持された入力データを確定済みの入力データとして送信する第１の入力候補調整制御ステップと、

前記入力候補収集ステップによる入力データの収集時に、他のコンピュータから確定済みの入力データが送信されたときに、その入力データを次に処理する対象として確定する第２の入力候補調整制御ステップと

をさらに具備することを特徴とする請求項６または７記載の分散システムの多重化制御方法。

【請求項８】　前記ジャーナルステップは、前記入力データを最新のものから予め定められた工程数だけ保持し、

前記第１の入力候補調整制御ステップは、他のコンピュータに送信すべき確定済みの入力データが前記ジャーナル手段に保持されていないときに、その旨を通知するステップを具備し、

自コンピュータで既に確定済みである各工程での直前の状態を予め定められた工程分まで保持する状態保持ステップと、

前記状態保持ステップにより保持された各工程での直前の状態を他のコンピュータとの間で送受信する状態送受信ステップと、

前記入力候補収集ステップによる入力データの収集時に、その収集された入力データ数と他のコンピュータから確定済みの入力データが前記ジャーナルステップによっても既に保持されていない旨を通知された数との和が（ｎ－ｆ）以上であって、前記収集された入力データ数が（ｎ－ｆ）未満であったときに、他のすべてのコンピュータの中で確定済みの工程が最も進んだ他のコンピュータにおける最新の確定済みの工程での直前の状態を前記状態送受信ステップにより取得して自コンピュータに複製するスキップステップと

をさらに具備することを特徴とする請求項７記載の分散システムの多重化制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は、４台以上のコンピュータがネットワークで接続された分散システムおよび同システムの多重化制御方法に係り、特に、スプリットブレインの防止と故障発生時におけるリアルタイム性の確保とを両立させることを可能とした分散システムおよび同システムの多重化制御方法に関する。

【０００２】

【従来の技術】

近年、コンピュータ技術やネットワーク技術の向上は目覚ましく、これに伴って、業務の電算化が広く行われている。また、その業務の内容によっては、故障

などによる中断が許されないものも多く、最近では、複数のコンピュータをネットワークで結合した分散システムを構築することが一般的になりつつある。そして、この分散システムの運用手法の１つに、整列マルチキャストを用いた決定性のプログラムの実行の多重化が存在する。

【０００３】

まず、「整列マルチキャスト」、「決定性のプログラム」および「多重化」について説明する。

【０００４】

・整列マルチキャスト

複数のコンピュータが結合した分散システムのような環境では、各コンピュータが独立して動作する。したがって、これらのコンピュータを同期的に動作させるためには、特別な仕組みが必要である。整列マルチキャストは、分散システムへの入力をすべてのコンピュータに配送する仕組みであり、データの到着順序がすべてのコンピュータで同じであることを保証するものである。

【０００５】

・決定性のプログラム

プログラムの実行は、コンピュータに入力が与えられると、その時のコンピュータの状態によって、出力と次の状態とを決めるものであると考えることができる。そして、決定性（deterministic）のプログラムは、与えられた入力にしたがって、出力と次の状態とが一意的に決まるプログラムとして定義される。具体的には、不定値や乱数の参照等がないプログラムのことをいう。決定性のプログラムの特徴は、初期状態と入力列とが決まれば、その実行が一意的であることである。以下、本明細書でプログラムと称するとき、決定性のプログラムのことをさすものとする。

【０００６】

・多重化

分散システムでは、各コンピュータが独立に故障する可能性がある。仮に、１つのコンピュータが故障しただけでシステム全体が機能しない場合は、分散システムの稼働率は、１台のコンピュータの稼働率よりも低くなってしまう。かかる

事態を防止するために、システム全体に係わる処理は多重化することが必要である。逆に、多重化することによって、分散システムの稼働率を１台のコンピュータの稼働率よりも高くすることが可能である。たとえば、稼働率９９パーセントのコンピュータ１０台で構成する分散システムが、まったく多重化されていないとすると、その分散システムの稼働率は９０％程度である。もし、これが多重化によって３台の故障まで耐え得るとすると、稼働率は、９９．９９９８％程度となる。

【０００７】

次に、整列マルチキャストを用いた決定性のプログラムの実行の多重化について説明する。ここでは、複数台のコンピュータによって構成される分散システムであって、多重化を構成するそれぞれのコンピュータが、同一のプログラムを有していると想定する。

【０００８】

まず、すべてのコンピュータは、同一の初期状態からはじまる。その後、入力されるデータは、必ず整列マルチキャストを通して、すべてのコンピュータに同一順序で配送され、それぞれのプログラムが実行される。

【０００９】

各プログラムへの入力列は、この整列マルチキャストにより、同一順序となっているので、決定性のプログラムの特徴により、すべてのコンピュータの状態が同一に保たれ、出力列もすべて同じとなる。つまり、プログラムの実行が多重化される。

【００１０】

ここで、整列マルチキャストの実現方法について、その概要を説明する。

【００１１】

特別なハードウェアによらずに整列マルチキャストを実現するためには、コンピュータ間で適切なアルゴリズムにしたがってメッセージをやり取りすること、つまりプロトコルが用いられる。アルゴリズムを具体的に説明する前に、注意すべき点を列挙する。

【００１２】

すべてのコンピュータが、いつでも故障停止する可能性があることを前提としており、多重化として成立するためには、特定のコンピュータに全体の処理が依存してはならない。したがって、次のことに注意する必要がある。

【００１３】

（１）分散システムへの入力の受け付けを特定のコンピュータに固定しない。

【００１４】

たとえば、特定のコンピュータに入力の受付を固定し、すべての入力をそのコンピュータにいったん転送することによって入力の順序を決定し、その順序で配送するといった単純なアルゴリズムは使えない。このアルゴリズムでは、入力受付を固定したコンピュータが故障停止すると、その時点で入力の順序が決定できなくなってしまう。

【００１５】

（２）入力の配送の完了まちあわせを特定のコンピュータに固定しない。

【００１６】

たとえば、特定のコンピュータが、停止していないすべてのコンピュータに配送を行うようにする、といった単純なアルゴリズムは使えない。このアルゴリズムでは、配送コンピュータが配送の途中で故障停止してしまうと、一部のコンピュータにのみ配送されたまま、配送が完了しなくなってしまう。

【００１７】

以上を踏まえて、前述のアルゴリズムを具体的に説明する。

【００１８】

従来では、故障検出が重要な役割を果たす。典型的には、故障検出はハートビート・タイムアウト・アルゴリズムによって行われる。このアルゴリズムは、各コンピュータが定期的に送出するハートビート（心拍）が一定時間以上確認できない場合に、当該コンピュータの故障を判定するというものである。

【００１９】

また、各コンピュータは、入力受付キューをもつ。第１ステップとして、それぞれのコンピュータは、入力受付キューの先頭にある入力をそのコンピュータにおける「入力候補」として他のすべてのコンピュータに配送する。また、入力受

付キューが空のコンピュータでは、他のコンピュータの第１ステップとして最初に得られた「入力候補」を自分の「入力候補」として他のすべてのコンピュータに配送する。

【００２０】

第１ステップの最終的な結果として、各コンピュータは、すべてのコンピュータについて、「入力候補」を得るか、「故障検出」を得るか、または、その双方を得る。ここでは、すべてのコンピュータについての「入力候補」および「故障検出」の一覧を単に「一覧」と呼ぶことにする。

【００２１】

第２ステップとして、それぞれのコンピュータは、自分の「一覧」を他のすべてのコンピュータに配送する。ここで注意すべき点は、これらの「一覧」が、各コンピュータごとに異なっているかも知れないということである。なぜなら、第１ステップの途中で故障停止が発生した場合には、「入力候補」が部分的にしか配送されていないかも知れない。また、第２ステップの開始の時点で、「故障検出」にはずれがあるかも知れないからである。

【００２２】

第２ステップの結果として、各コンピュータは、他のコンピュータから得られた「一覧」が自分の「一覧」と異なっている場合、これらを合併して自分の「一覧」にし、第２ステップを繰り返し実行する。すると、この第２ステップの最終的な結果として、故障していない他のコンピュータがもつ「一覧」がすべて自分の「一覧」と一致する。この時点で、プロトコルは完了する。

【００２３】

なお、整列マルチキャストとして配送される入力は、その「一覧」にある「入力候補」の中から各自が同一の決まったルールで選べばよい（たとえば先頭にあるもの）。そして、最後に、その選んだ入力を入力受付キューから取り除く。

【００２４】

以上の手順により、複数のコンピュータをネットワークで結合した分散システムにおける、整列マルチキャストを用いた決定性のプログラムの実行の多重化が実現される。

【００２５】

【発明が解決しようとする課題】

ところで、前述した手順では、次のような問題点があった。

【００２６】

（１）スプリットブレイン

スプリットブレインは、実行のコンテキスト（状態）が２つ以上に分かれてしまうことをさす。このスプリットブレインは、故障検出が誤って行われたときに発生する。たとえば、システムを構成するコンピュータが、２つのコンピュータ群の間で互いに通信できない状態となった場合（ネットワークパーティショニング）、それぞれのコンピュータ群は、互いに故障検出し、独立して動作をはじめる。あるいは、一時的な高負荷のために、ハートビートの送受信が中断して故障の誤検出が発生し、スプリットブレインに陥る場合もある。

【００２７】

多重化された処理は、システムの中で重要な処理であるはずである。ここでスプリットブレインが起きると、その処理に一貫性がなくなり、システム全体に致命的な影響を及ぼすことになる。

【００２８】

スプリットブレインを起きにくくするためには、故障の誤検出を起きにくくする必要がある。そのためには、ハートビートのタイムアウトを十分に長くする必要がある。実用上は、１０秒～１分ぐらいのタイムアウト値が使われるのが一般的である。

【００２９】

（２）故障発生時の処理のリアルタイム性

ところが、タイムアウトを長く設定すると、故障の発生から故障検出までの時間が長くなることになる。すると、その間は、整列マルチキャストのプロトコル中で、故障したコンピュータの故障検出を待ち、整列マルチキャストの実行が一時的に停止する。その結果、多重化の実行が一時的に停止することになる。

【００３０】

これは、一般的にはシステムに致命的な影響を与えるものではないが、リアル

タイム性が重要なシステムでは、故障発生時にその要件を満たさなくなる場合もある。つまり、ハートビートのタイムアウト値は、リアルタイム性の要件から上限が抑えられており、むやみに長く設定できない。

【００３１】

結局、このハートビートのタイムアウト値の設定は、スプリットブレインとリアルタイム性の間でトレードオフの関係に陥ってしまうという問題があった。

【００３２】

この発明は、このような事情を考慮してなされたものであり、スプリットブレインの防止と故障発生時におけるリアルタイム性の確保とを両立させることを可能とした分散システムおよび同システムの多重化制御方法を提供することを目的とする。

【００３３】

【課題を解決するための手段】

前述した目的を達成するために、この発明は、故障検出をまったく行わないことによって、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させないようにしたものである。そして、そのために、この発明は、少なくとも（ｎ－ｆ）台のコンピュータが動作していれば、他のｆ台の動作に関わらず、入力をそれらに配送するようにした。

【００３４】

より具体的には、この発明は、ネットワークで接続されたｎ台のコンピュータを同期的に動作させる分散システムであって、少なくとも（ｎ－ｆ）台以上での多重化を保証する分散システムにおいて、前記コンピュータそれぞれは、各々が次に処理する候補として選択した入力データを前記ネットワークを介して収集する入力候補収集手段と、前記入力候補収集手段により収集された入力データが（ｎ－ｆ）個以上存在する場合に、その中に同一内容の入力データが（ｎ－ｆ）個以上あるか否かを判定し、（ｎ－ｆ）個以上あったときに、その入力データを次に処理する対象として確定する第１の入力候補選定制御手段と、前記収集された入力データの中に同一内容の入力データが（ｎ－ｆ）個以上なかったときに、前記収集された入力データ数の過半数を占める同一内容の入力データが存在するか

否かを判定し、存在したときに、その入力データを自候補とするとともにそれ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第２の入力候補選定制御手段と、前記収集された入力データ数の過半数を占める同一内容の入力データが存在しなかったときに、前記収集された入力データの中からいずれかの入力データを任意に選択して自候補とするとともに、それ以外の他の候補の入力データをすべて破棄した上で前記入力候補収集手段に入力データの収集を再実行させる第３の入力候補選定制御手段とを具備することを特徴とする分散システムを提供する。

【００３５】

この分散システムにおいては、故障検出をまったく行わないで、整列マルチキャストを実現し、特に、故障発生時でも配送の中断を発生させることがない。

【００３６】

【発明の実施の形態】

以下、図面を参照してこの発明の一実施形態を説明する。

【００３７】

まず、この実施形態に係る分散システムの前提条件を説明する。ここでは、多重化を構成するコンピュータの数をｎとし、ｆ台までの故障停止が許容されるものと想定する。つまり、多重化されるプログラムは、少なくとも（ｎ－ｆ）台のコンピュータ上で実行される。また、（ｆ＋１）台以上の故障停止が発生した場合には、多重化は継続しないものとする（いわゆるフェイルストップ）。

【００３８】

また、ここでは、このｆを３ｆ＜ｎとなる最大の整数とする。たとえばｎ＝４ならば、ｆ＝１である。ｎ＝１０ならばｆ＝３である。この前提は、システムの稼働率に制限を設けるものであるが、たとえば、ｎ＝１０の場合には、前述の稼働率の計算より、実用上まったく問題ないといえる。

【００３９】

また、多重化されるプログラムの入力および出力は、信頼性のないデータグラム（Unreliable Datagram）のセマンティックスであるとする。これは、入出力のパケットについて、欠落、重複および順序の交換を許容するものである。信頼

性のないデータグラムのセマンティクスをもつ例としては、ＩＰ（Internet Protocl）が挙げられる。

【００４０】

この信頼性のないデータグラムのセマインティクスにおける非決定性と多重化されるプログラムの決定性とは矛盾するものではないことに注意する。プログラムの決定性は、入力が決まれば一意的に次の状態と出力が決まることを示し、プログラムの内部動作に関する決定性を意味している。一方、信頼性のないデータグラムのセマンティクスは、あるプログラムの出力が他のプログラムの入力へ渡される途中で、欠落、重複または順序の交換があり得ることを示し、プログラム間の入出力に関する非決定性を意味している。

【００４１】

次に、図１および図２を参照して、この分散システムの構成を説明する。

【００４２】

図１に示すように、この分散システム１０００は、ｎ台のコンピュータ１００により多重化されており、それぞれのコンピュータ１００が、外部ネットワークＡを介して複数のクライアント装置２０００と接続されている。また、このコンピュータ１００間は、内部ネットワークＢを介して接続されている。そして、この分散システム１０００における各コンピュータ１００は、外部ネットワークＡを介してクライアント装置２０００から受け取った入力パケット（入力１）または内部ネットワークＢを介して他のコンピュータ１００から受け取った入力パケット（入力２）を他のコンピュータ１００と同じ順序で処理していく。

【００４３】

なお、この処理により生成される出力パケットは、外部ネットワークＡを介してクライアント装置２０００に返却され（出力１）、または、内部ネットワークＢを介して他のコンピュータ１００に転送される。

【００４４】

図２は、コンピュータ１００の構成を示す図である。入力受付キュー部１で受け付けられた入力パケットは、整列マルチキャスト部２によってアプリケーションプログラム３に配送されることになる。配送された入力パケットの入力によっ

て、このアプリケーションプログラム３は、プログラム状態管理部４に保存されている状態にしたがって実行し、出力パケットを生成する。出力パケットは、出力フィルター部５で選別されてから出力される。

【００４５】

次に、整列マルチキャスト部２の各構成要素について説明する。

【００４６】

入力順序番号記憶部２１は、整列マルチキャストによってそのコンピュータへ次に配送される入力パケットの順序番号を記憶する。入力パケットジャーナル記憶部２２は、整列マルチキャストによってそのコンピュータへ配送が確定した入力パケットの列を最近のものから一定の量だけ記憶する。プロトコルデータ送受信部２３は、他のコンピュータのプロトコルデータ送受信部２３とプロトコルデータをやり取りする。

【００４７】

また、ステップ番号記憶部２４、候補パケット記憶部２５および入力パケット確定判定部２６は、整列マルチキャストによってそのコンピュータへ次に配送される入力パケットを決定するアルゴリズムで用いられる。ステップ番号記憶部２４は、プロトコルのステップ番号を記憶する。候補パケット記憶部２５は、そのステップにおける各コンピュータの「入力候補」となる入力パケットを計ｎ個記憶する。入力パケット確定判定部２６は、候補パケット記憶部２５の情報から入力パケットの確定の判定および次ステップの「入力候補」の決定を行う。

【００４８】

最大確定入力順序番号記憶部２７は、他のコンピュータも含め、配送が確定したことがわかっている最大の入力順序番号を記憶する。遅延記憶部２８は、（ｎ－１）個のフラグで構成され、他コンピュータから遅延しているかどうかを記憶する。そして、スキップ判定部２９は、遅延記憶部２８の情報からスキップ動作の必要性を判定、実行する。

【００４９】

以降、該当入力順序番号とは、入力順序番号記憶部２１に記憶された入力順序番号のことを指し、該当ステップ番号とは、ステップ番号記憶部２４に記憶され

たステップ番号のことを指し、該当最大確定入力順序番号とは、最大確定入力順序番号記憶部２７に記憶された入力順序番号のことを指し、自候補とは、候補パケット記憶部２５における自コンピュータに対応する「入力候補」を指し、他候補とは、候補パケット記憶部２５における自候補以外の「入力候補」を指すものとする。

【００５０】

図３は、プロトコルデータ送受信部２３によって送受信されるプロトコルデータのレイアウトを示す図である。

【００５１】

図３に示すように、プロトコルデータ送受信部２３によって送受信されるプロトコルデータは、種類、送信者、入力順序番号、ステップ番号、最大確定入力順序番号および入力パケットの各フィールドを含んでいる。そして、先頭の種類フィールドによって、このプロトコルデータは、次の３つに使い分けられる。

【００５２】

（１）候補種類：入力順序番号フィールド、ステップ番号フィールド、入力パケットフィールドには、それぞれ、送信者の送信時における、該当入力順序番号、該当ステップ番号、自候補が格納される。

【００５３】

（２）確定種類：その入力順序番号に対応する入力パケットが、送信者の送信時における入力パケットジャーナル記憶部２６にあることを示し、入力パケットフィールドにはその入力パケットが格納される。この場合、ステップ番号フィールドは使用しない。

【００５４】

（３）遅延種類：その入力順序番号に対応する入力パケットが、送信者の送信時における入力パケットジャーナル記憶部２６にないことを示す。この場合、ステップ番号フィールド、入力パケットフィールドは使用しない。

【００５５】

いずれの種類においても、最大確定入力順序番号フィールドには、送信者の送信時における該当最大確定入力順序番号を格納する。また、該当最大確定入力順

序番号は、そのコンピュータで確定した入力パケットの順序番号と、受信したプロトコルデータ中の最大確定入力順序番号とのうち、最も大きいものに更新するものとする。

【００５６】

ここで、図４を参照して、整列マルチキャスト部２によって実行される整列マルチキャストの主要部の概要について説明する。

【００５７】

いま、多重化を構成するコンピュータの数、つまりｎを４とする。また、前述したように、ｆは３ｆ＜ｎとなる最大の整数であるから、ｆ＝１となる。したがって、この例では、少なくとも（ｎ－ｆ）、つまり３台以上で一貫性を保ちながら処理を実行していくことになる。

【００５８】

第１に、コンピュータ（１），（２）はＡ、コンピュータ（３）はＢ、コンピュータ（４）はＣをそれぞれ入力候補として選択したとする。また、第２に、コンピュータ（１）は、コンピュータ（２）の入力候補Ａとコンピュータ（３）の入力候補Ｂを収集したとする。つまり、コンピュータ（１）は、自候補および他候補を（ｎ－ｆ）個収集したことになる。この時、コンピュータ（１）は、コンピュータ（４）の入力候補の収集を待たずに、入力候補の判定を試みる。しかしながら、その中に（ｎ－ｆ）個の同一の候補は存在しないことから、コンピュータ（１）は、入力候補の再選択を実行する。再選択は、収集された入力候補数の過半数を占める候補があればその候補を選択し、なければその中からランダムに選択する。ここではＡが過半数を占めるので、コンピュータ（１）は、第３に、Ａを自候補として再選択する。

【００５９】

この要領で、コンピュータ（２）は、コンピュータ（１）の入力候補Ａとコンピュータ（４）の入力候補Ｃを収集した後、Ａを自候補として再選択し、コンピュータ（３）は、コンピュータ（２）の入力候補Ａとコンピュータ（４）の入力候補Ｃを収集した後、Ｃを自候補として再選択し、コンピュータ（４）は、コンピュータ（１）の入力候補Ａとコンピュータ（２）の入力候補Ａを収集した後、

Ａを自候補として再選択したとする。

【００６０】

第４に、コンピュータ（１）は、コンピュータ（２）の入力候補Ａとコンピュータ（４）の入力候補Ａを収集したとする。つまり、コンピュータ（１）は、再度、自候補および他候補を（ｎ－ｆ）個収集したことになる。この時、コンピュータ（１）は、コンピュータ（３）の入力候補の収集を待たずに、入力候補の判定を試みる。そして、ここでは、（ｎ－ｆ）個のＡが存在するため、第５に、コンピュータ（１）は、入力をＡに決定する。

【００６１】

一方、コンピュータ（２）は、コンピュータ（１）の入力候補Ａとコンピュータ（３）の入力候補Ｃを収集したとする。しかしながら、（ｎ－ｆ）個の同一の候補は依然として存在しないことから、コンピュータ（２）は、入力候補の再選択を実行し、その中の過半数を占めるＡを自候補として選択する。同様に、コンピュータ（３）は、コンピュータ（１）の入力候補Ａとコンピュータ（２）の入力候補Ａを収集した後、コンピュータ（４）は、コンピュータ（２）の入力候補Ａとコンピュータ（３）の入力候補Ａを収集した後、それぞれＡを自候補として再選択したとする。

【００６２】

第６に、コンピュータ（２）は、コンピュータ（１）の入力候補Ａとコンピュータ（３）の入力候補Ａを収集したとする。ここでのコンピュータ（１）の入力候補Ａは、既に候補ではなく確定済みの入力であるため、第７に、コンピュータ（２）は、入力をＡに決定する。

【００６３】

一方、コンピュータ（３）は、コンピュータ（２）の入力候補Ａとコンピュータ（４）の入力候補Ａを収集し、コンピュータ（４）は、コンピュータ（２）の入力候補Ａとコンピュータ（３）の入力候補Ａを収集したとする。そして、ここでは、双方とも（ｎ－ｆ）個のＡが存在するため、コンピュータ（３），（４）は、入力をＡに決定する。

【００６４】

つまり、この分散システムは、従来のように、各コンピュータがハートビートのやり取りによって他のコンピュータとの間で正常稼働を確認し合うようなことを一切行わないことにより、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させないようにし、かつ、少なくとも（ｎ－ｆ）台以上のコンピュータによる多重化を保証する。

【００６５】

次に、整列マルチキャスト部２の動作原理について具体的に説明する。

【００６６】

まず、初期状態として、入力順序番号記憶部２１は初期入力順序番号（たとえば１）を記憶する。入力パケットジャーナル記憶部２２は空の状態であり、ステップ番号記憶部２４は初期ステップ番号（たとえば１）を記憶する。また、候補パケット記憶部２５も空の状態であり、最大確定入力順序番号記憶部２７は初期入力順序番号を記憶し、さらに、遅延記憶部２８のすべてのフラグはリセットされている。

【００６７】

そして、この整列マルチキャスト部２が実行する整列マルチキャストによって各コンピュータへ配送される入力パケットを決定するアルゴリズムの概要は次のようになる。

【００６８】

（アルゴリズム１）

該当ステップ番号が初期ステップ番号である場合に、入力受付キュー部１に入力パケットがあれば、該当ステップ番号を次に進め、自候補をその入力パケットにし、他候補を空にし、候補種類のプロトコルデータを他のすべてのコンピュータに送信する。

【００６９】

（アルゴリズム２）

該当入力順序番号に一致する入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、そのプロトコルデータが該当ステップ番号より大きいステップ番号を持つ場合、該当ステップ番号をそのステップ番号にし、自候補および送

信者に対応する他候補をプロトコルデータ中の入力パケットにし、それら以外の他候補を空にし、候補種類のプロトコルデータを他のすべてのコンピュータに送信する。

【００７０】

（アルゴリズム３）

該当入力順序番号に一致する入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、そのプロトコルデータが該当ステップ番号と等しいステップ番号を持つ場合、送信者に対応する他候補をプロトコルデータ中の入力パケットにする。

【００７１】

（アルゴリズム４）

候補パケット記憶部２５における空でない「入力候補」が（ｎ－ｆ）個以上あるとき、入力パケット確定判定部２６は次の動作をする。

【００７２】

もし、（ｎ－ｆ）個以上の同一内容の「入力候補」があれば、それを該当入力順序番号における入力パケットとして確定し、入力パケットジャーナル記憶部２２に記憶し、入力受付キュー部１にそれがあれば削除し、アプリケーションプログラム３に配送し、該当入力順序番号を次に進め、該当ステップ番号を初期ステップ番号にし、候補パケット記憶部２５を空にし、遅延記憶部２８のすべてのフラグをリセットする。

【００７３】

それ以外で、もし、候補パケット記憶部２５の中で過半数以上の同一内容の「入力候補」があれば、該当ステップ番号を次に進め、候補パケット記憶部２５における自候補をその入力パケットにし、他候補を空にし、候補種類のプロトコルデータを他のすべてのコンピュータに送信する。

【００７４】

さらに、それ以外であれば、候補パケット記憶部２５の中からランダムに入力パケットを選択し、該当ステップ番号を次に進め、候補パケット記憶部２５における自候補をその入力パケットにし、他候補を空にし、候補種類のプロトコルデ

ータを他のすべてのコンピュータに送信する。

【００７５】

（アルゴリズム５）

該当入力順序番号より小さい入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、その入力順序番号に対応する入力データが入力パケットジャーナル記憶部２２にある場合、確定種類のプロトコルデータを送信者のコンピュータに返信する。

【００７６】

（アルゴリズム６）

該当入力順序番号に一致する入力順序番号を持つ確定種類のプロトコルデータを受信した場合、それを該当入力順序番号における入力パケットとして確定し、入力パケットジャーナル記憶部２６に記憶し、入力受付キュー部１にそれがあれば削除し、アプリケーションプログラム３に配送し、該当入力順序番号を次に進め、該当ステップ番号を初期ステップ番号にし、候補パケット記憶部を空にし、遅延記憶部２８のすべてのフラグをリセットする。

【００７７】

（アルゴリズム７）

該当入力順序番号より小さい入力順序番号を持つ候補種類のプロトコルデータを受信した場合で、その入力順序番号に対応する入力データが入力パケットジャーナル記憶部２２にない場合、遅延種類のプロトコルデータを送信者のコンピュータに返信する。

【００７８】

（アルゴリズム８）

該当入力順序番号に一致する入力順序番号を持つ遅延種類のプロトコルデータを受信した場合に、遅延記憶部２８における送信者に対応するフラグをセットする。

【００７９】

（アルゴリズム９）

遅延記憶部２８においてフラグがたっている数と、それ以外で候補パケット記

憶部２５における空でない入力候補数の和が（ｎ－ｆ）以上であるときで、候補パケット記憶部２５における空でない入力候補数が（ｎ－ｆ）個未満であるときに、スキップ判定部２９は、以下のスキップ動作を行う。

【００８０】

スキップ動作は、該当入力順序番号を該当最大確定入力順序番号にし、該当ステップ番号を初期ステップ番号にし、候補パケット記憶部２５を空にし、遅延記憶部２８のすべてのフラグをリセットし、プログラム状態管理部４にスキップを通知する。

【００８１】

なお、以上の（アルゴリズム１）～（アルゴリズム９）の順序は、必ずしもこの順序で実行されるというものではない。つまり、これらは、その条件が成立すれば独立して実行されるものである。

【００８２】

また、プログラム状態管理部４は、スキップが通知されると、該当入力順序番号の直前の状態を他のコンピュータのプログラム状態管理部４からコピーする。このために、プログラム状態管理部４は、各入力順序番号の直前の状態を最近のものから一定の量だけ保持している。

【００８３】

ここで、上述したアルゴリズムの動作の概要を説明しながら、このアルゴリズムの有効性を証明する。

【００８４】

（アルゴリズム１）～（アルゴリズム４）は、整列マルチキャストの１回の配送を行う基本的な部分である。従来では、故障していない全コンピュータで一致するまで繰り返していたのに対して、この分散システムでは、（ｎ－ｆ）台で一致するまで繰り返す。

【００８５】

また、（アルゴリズム５）～（アルゴリズム６）は、短い多重化実行の遅延を解消するため、すでに確定している入力パケットを回送するものである。

【００８６】

そして、（アルゴリズム７）～（アルゴリズム９）は、長い多重化実行の遅延を一足飛びに解消するため、スキップ動作を行うものである。

【００８７】

まず、（アルゴリズム１）～（アルゴリズム６）が整列マルチキャストの要件を満たすことを説明する。これには、各入力順序番号で同一の入力パケットが確定されることを示せばよい。

【００８８】

入力パケットを確定するのは、（アルゴリズム４）か（アルゴリズム６）であるが、（アルゴリズム６）の場合は、確定した入力パケットを回送したものなので、最初に（アルゴリズム４）によって入力パケットを確定したコンピュータが必ず存在する。確定した時の入力パケットをＰ、ステップ番号をＳとする。

【００８９】

まず、ステップＳ＋１では、すべてのコンピュータで「入力候補」はＰ以外にはあり得ないことを示す。

【００９０】

自分の「入力候補」を決定するのは、（アルゴリズム１）、（アルゴリズム２）または（アルゴリズム４）であるが、ステップ番号Ｓは初期ステップ番号ではあり得ないので、ステップＳ＋１での「入力候補」は、（アルゴリズム２）か（アルゴリズム４）で決定される。（アルゴリズム２）は「入力候補」を回送したものなので、結局、（アルゴリズム４）で決定するステップＳ＋１での「入力候補」がＰ以外にはあり得ないことを示せばよい。

【００９１】

ステップＳ＋１での「入力候補」を（アルゴリズム４）で決定するには、ステップＳでの「入力候補」が（ｎ－ｆ）個必要である。この集合をＸとする。一方、ステップＳでは、（アルゴリズム４）によって入力パケットを確定したコンピュータがあるのだから、少なくとも（ｎ－ｆ）個の「入力候補」がＰである。この集合をＹとする。すると、

Ｘの要素数≧ｎ－ｆ

Ｙの要素数≧ｎ－ｆ

ＸＵＹの要素数≦ｎ

Ｘの要素数－Ｘ∩Ｙの要素数＝ＸＵＹの要素数－Ｙの要素数≦ｎ－（ｎ－ｆ）＝ｆ

となり、ＸのうちＰでないのは、多くともｆ個しかない。後は、ｆがＸの中で半数未満であることがいえれば、Ｘの中でＰが過半数を占めることになり、（アルゴリズム４）によってＰに決定することがわかる。ここで、

Ｘの要素数－２ｆ≧（ｎ－ｆ）－２ｆ＝ｎ－３ｆ

となり、前述の通り、ｎ－３ｆ＞０であるから、これが証明される。

【００９２】

結局、ステップＳ＋１では、すべてのコンピュータで「入力候補」はＰ以外にはあり得ないのだから、この入力順序番号で確定するとすれば、必ずＰで確定することになる。これで、整列マルチキャストの要件を満たすことが言えた。

【００９３】

次に、（アルゴリズム５）～（アルゴリズム９）で行う遅延の解消について説明する。

【００９４】

この遅延は、（ｎ－ｆ）台よりも多い台数で多重化を実行している場合に発生する。遅延しているコンピュータは、その時点では多重化として不要であるが、進んでいるコンピュータが故障停止した場合などに、多重化を継続するために必要になる。つまり、その場合には、遅延しているコンピュータは、最終入力順序番号まで追いつかなければならない。

【００９５】

（アルゴリズム５）～（アルゴリズム６）で行う短い多重化実行の遅延の解消は、単純に、進んでいるコンピュータで確定した入力パケットを回送する。入力パケットの到着順序は同じになるので、整列マルチキャストの要件は満たされている。

【００９６】

一方、（アルゴリズム７）～（アルゴリズム９）で行う長い多重化実行の遅延の解消は、いわゆる「おいてけぼり」の概念を用いる。「おいてけぼり」は、進

んでいるコンピュータが確定した入力パケットを忘れてしまうほど長く遅延したときに発生する。そして、この「おいてけぼり」が判定されると、スキップ動作が行われる。スキップ動作では、入力順序番号をスキップするので、入力パケットの系列が中抜けになり、整列マルチキャストの要件を満たさなくなる。

【００９７】

そこで、この中抜けになった入力パケットの系列を補うため、プログラム状態管理部４により一致化コピーを行う。これによって、多重化は矛盾なく続行することができる。

【００９８】

次に、信頼性のないデータグラムのセマンティクスとの関係に触れる。

【００９９】

出力に関しては、信頼性のないデータグラムのセマンティクスなので、出力フィルタ部５の動作は任意でよい。たとえば、無選別で出力すると、出力パケットが多重化を実行するコンピュータの数だけ出力されることになるが、信頼性のないデータグラムのセマンティクスでは、パケットの重複を許すので、この範囲内である。

【０１００】

また、この分散システムでは、多重化実行の遅延が発生するため、特に出力パケットに関して順序の交換が発生する可能性がある。これは、進んでいるコンピュータが出力した後、遅延しているコンピュータが意味的にはそれ以前の出力を実行するためである。

【０１０１】

しかしながら、性能面などにおいて、出力フィルタ部５の設定は重要であり、たとえば、（アルゴリズム４）で入力パケットが確定したときは、出力フィルタを開、（アルゴリズム６）で入力パケットが確定したときは、出力フィルタを閉と設定すれば、順序の交換を低減することができる。また、（アルゴリズム４）で入力パケットが確定し、その入力パケットが入力受付キュー１から取り除かれた場合にのみ、出力フィルタを開、それ以外では閉とすれば、重複を低減することができる。

【０１０２】

すなわち、この分散システムは、すくなくともｎ－ｆ台のコンピュータが動作していれば、他のｆ台の動作に関係なく入力をそれらに配送することにより、整列マルチキャストを故障検出を使わないで実現し、特に、故障発生時でも、配送の中断が発生しない。

【０１０３】

また、最大でｆ台のコンピュータで、プログラムの多重化の実行が遅延する可能性があることを考慮し、この遅延された実行がスプリットブレインを起こさないように追い付く仕組みを実現する。

【０１０４】

次に、図５乃至図１０を参照して、整列マルチキャスト部２の動作手順について説明する。

【０１０５】

図５および図６は、整列マルチキャストの１回の配送を行う基本的な部分の動作手順を示すフローチャートである。

【０１０６】

整列マルチキャスト部２は、まず、候補一覧作成処理を実行する（図５のステップＡ１）。この候補一覧作成は、該当ステップ番号が初期値のときは（図６のステップＢ１のＹＥＳ）、受付キューに入力パケットが存在するかどうかを調べて（図６のステップＢ２）、存在すれば（図６のステップＢ２のＹＥＳ）、該当ステップ番号を次に進め（図６のステップＢ３）、受付キューの入力パケットを自候補とし、かつ、この自候補を他のすべてのコンピュータに送信する（図６のステップＢ４）。

【０１０７】

一方、該当ステップ番号が初期値でないか（図６のステップＢ１のＮＯ）、または受付キューに入力パケットがないとき（図６のステップＢ２のＮＯ）、整列マルチキャスト部２は、同一の入力順序番号を持つプロトコルデータを受信しているかどうか判定し（図６のステップＢ５）、受信していれば（図６のステップＢ５のＹＥＳ）、今度は、受信データ内のステップ番号は該当ステップ番号より

も大きいかどうかを判定する（図６のステップＢ６）。そして、該当ステップ番号よりも大きければ（図６のステップＢ６のＹＥＳ）、整列マルチキャスト部２は、該当ステップを受信データ内のステップ番号に更新した後（図６のステップＢ７）、受信データ内の入力パケットを自候補とし、かつ、この自候補を他のすべてのコンピュータに送信する（図６のステップＢ８）。このとき、整列マルチキャスト部２は、この入力パケットを他候補として記憶しておく。また、受信データ内のステップ番号と該当ステップ番号とが等しければ（図６のステップＢ６のＮＯ，ステップＢ９のＹＥＳ）、受信データ内の入力パケットを他候補として記憶する（図６のステップＢ１０）。

【０１０８】

ここで、整列マルチキャスト部２は、記憶した候補数が（ｎ－ｆ）個以上になったかどうかを調べ（図６のステップＢ１１）、なっていなければ（図６のステップＢ１１のＮＯ）、ステップＢ１からの処理を繰り返し、なっていれば（図６のステップＢ１１のＹＥＳ）、この処理を終了する。

【０１０９】

候補一覧作成処理が終了すると、整列マルチキャスト部２は、（ｎ－ｆ）個以上の同一の候補が存在するかどうかを調べ（図５のステップＡ２）、存在すれば（図５のステップＡ２のＹＥＳ）、その候補を入力パケットとして確定する（図５のステップＡ３）。つまり、この入力パケットを受付キューから削除するとともに、アプリケーションプログラム３に投入する。そして、整列マルチキャスト部２は、次工程へ移行すべく、入力順序番号を次に進め、該当ステップ番号を初期化し、記憶したすべての候補を破棄し、遅延フラグをリセットする（図５のステップＡ４）。

【０１１０】

一方、（ｎ－ｆ）個以上の同一の候補が存在しなかった場合（図５のステップＡ２）、整列マルチキャスト部２は、今度は、過半数以上の同一の候補が存在するかどうかを調べ（図５のステップＡ５のＹＥＳ）、存在すれば（図５のステップＡ５のＹＥＳ）、その候補を自候補とし、かつ、この自候補を他のすべてのコンピュータに送信した上で（図５のステップＡ６）、ステップＡ１からの処理を

繰り返す。この時、整列マルチキャスト部２は、記憶していた他候補をすべて破棄する。また、過半数以上の同一の候補が存在しなければ（図５のステップＡ５のＮＯ）、整列マルチキャスト部２は、ランダムに自候補を選択し、かつ、この自候補を他のすべてのコンピュータに送信した上で（図５のステップＡ７）、ステップＡ１からの処理を繰り返す。この時も、整列マルチキャスト部２は、記憶していた他候補をすべて破棄する。

【０１１１】

以上の手順で、各コンピュータは、故障検知を行わず、（ｎ－ｆ）台以上の一致を確認しながら処理を進めていく。

【０１１２】

また、図７乃至図１０は、多重化実行の遅延を解消するための動作手順を示すフローチャートである。

【０１１３】

整列マルチキャスト部２は、該当入力順序番号より小さい入力順序番号を持つ候補種類のプロトコルデータを受信した場合、その入力順序番号に対応する入力パケットがジャーナルに存在するかどうかを調べる（図７のステップＣ１）。そして、整列マルチキャスト部２は、ジャーナルに存在すれば（図７のステップＣ１のＹＥＳ）、その入力パケットをセットした確定種類のプロトコルデータを送信者に返送し（図７のステップＣ２）、一方、存在しなければ（図７のステップＣ１のＮＯ）、遅延種類のプロトコルデータを送信者に返送する（図７のステップＣ３）。

【０１１４】

また、整列マルチキャスト部２は、該当入力順序番号に一致する入力番号を持つ確定種類のプロトコルデータを受信した場合、その受信データ内の入力パケットを入力パケットとして確定する（図８のステップＤ１）。つまり、この入力パケットを受付キューから削除するとともに、アプリケーションプログラム３に投入する。そして、整列マルチキャスト部２は、次工程へ移行すべく、入力順序番号を次に進め、該当ステップ番号を初期化し、記憶したすべての候補を破棄し、遅延フラグをリセットする（図８のステップＤ２）。

【０１１５】

また、整列マルチキャスト部２は、該当入力順序番号に一致する入力順序番号を持つ遅延種類のプロトコルデータを受信した場合、送信者に対応する遅延フラグをセットする。

【０１１６】

また、整列マルチキャスト部２は、セットされた遅延フラグ数と記憶された候補数との和が（ｎ－ｆ）個以上に達したかどうかを監視し（図１０のステップＦ１）、（ｎ－ｆ）個以上に達していれば（図１０のステップＦ１のＹＥＳ）、その記憶された候補数が（ｎ－ｆ）個未満かどうかを調べる（図１０のステップＦ２）。そして、（ｎ－ｆ）個未満であれば（図１０のステップＦ２のＹＥＳ）、整列マルチキャスト部２は、スキップ動作を行なう（図１０のステップＦ３）。つまり、該当入力順序番号を該当最大確定入力順序番号にし、該当ステップ番号を初期ステップ番号にし、候補パケット記憶部２５を空にし、遅延記憶部２８のすべてのフラグをリセットした上で、プログラム状態管理部４にスキップを通知する。

【０１１７】

以上の手順で、各コンピュータは、スプリットブレインを起こさないよう、遅延された実行が追い付く仕組みを実現する。

【０１１８】

なお、本発明は、上記実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。更に、上記実施形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組み合わせにより種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題が解決でき、発明の効果の欄で述べられている効果が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【０１１９】

【発明の効果】

以上、詳述したように、この発明によれば、ｎ台のコンピュータで多重化を構

成し、ｆ台までの故障停止が許容される場合に、少なくとも（ｎ－ｆ）台のコンピュータが動作していれば、他のｆ台の動作に関わらず、入力がそれらに配送されるようになる。つまり、故障検出をまったく行わないことによって、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させることがない。

【０１２０】

また、最大でｆ台のコンピュータで、プログラムの多重化の実行が遅延する可能性があることを考慮し、この遅延された実行がスプリットブレインを起こさないように追い付く仕組みも実現する。

【図面の簡単な説明】

【図１】

この発明の実施形態に係る分散システムの構成を示す図。

【図２】

同実施形態の分散システムを構成するコンピュータの機能ブロック図。

【図３】

同実施形態の分散システムを構成するコンピュータ間で送受信されるプロトコルデータのレイアウトを示す図。

【図４】

同実施形態の分散システムが実行する整列マルチキャストの主要部の概要について説明するための図。

【図５】

同実施形態の分散システムが実行する整列マルチキャストの１回の配送を行う基本的な部分の動作手順を示す第１のフローチャート。

【図６】

同実施形態の分散システムが実行する整列マルチキャストの１回の配送を行う基本的な部分の動作手順を示す第２のフローチャート。

【図７】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第１のフローチャート。

【図８】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第２のフローチャート。

【図９】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第３のフローチャート。

【図１０】

同実施形態の分散システムが実行する、多重化実行の遅延を解消するための動作手順を示す第４のフローチャート。

【符号の説明】

１…入力受付キュー

２…整列マルチキャスト部

３…アプリケーションプログラム

４…プログラム状態管理部

５…出力フィルタ部

２１…入力順序番号記憶部

２２…入力パケットジャーナル記憶部

２３…プロトコルデータ送受信部

２４…ステップ番号記憶部

２５…候補パケット記憶部

２６…入力パケット確定判定部

２７…最大確定入力順序番号記憶部

２８…遅延記憶部

２９…スキップ判定部

１００…コンピュータ

１０００…分散システム

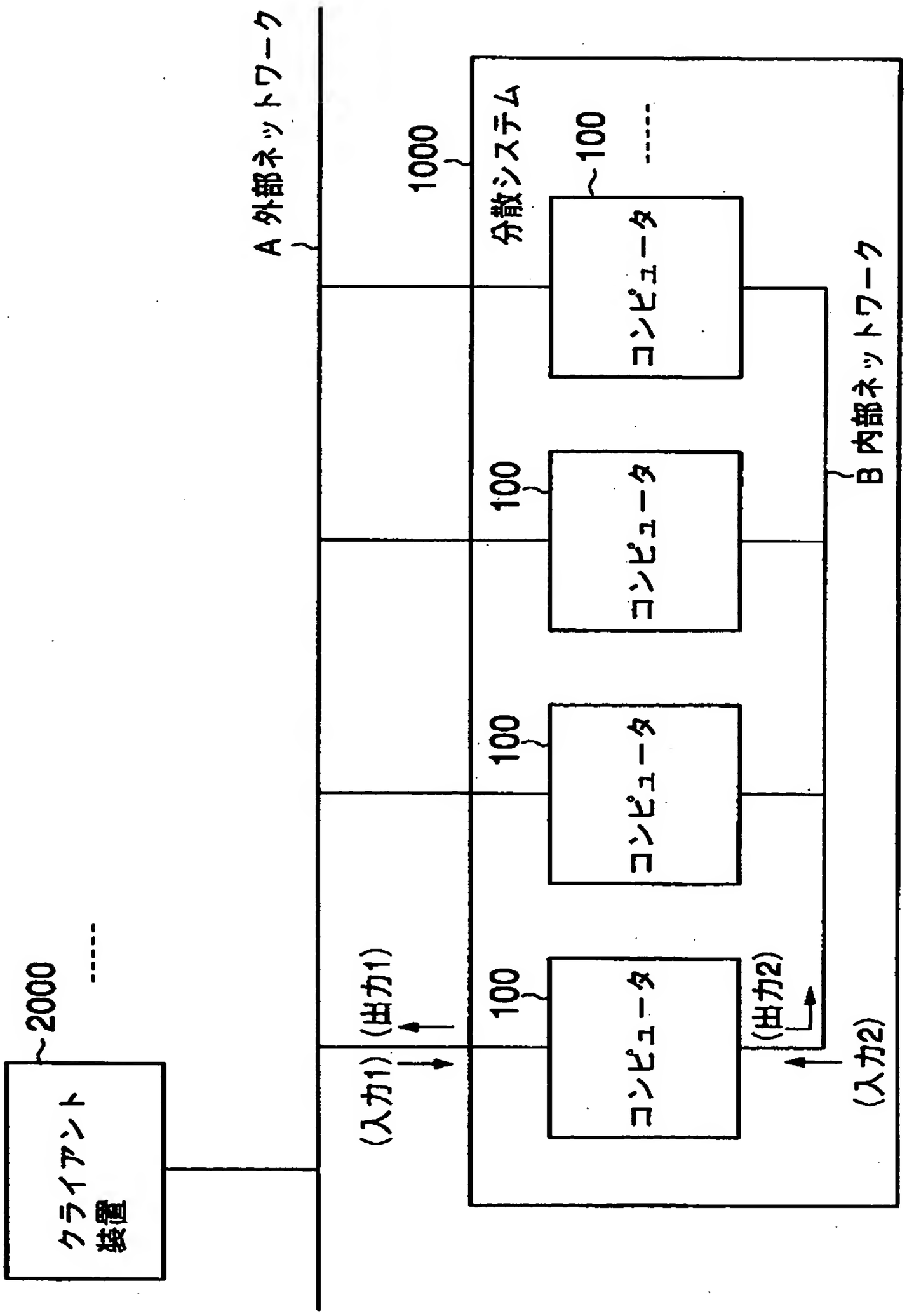
２０００…クライアント装置

Ａ…外部ネットワーク

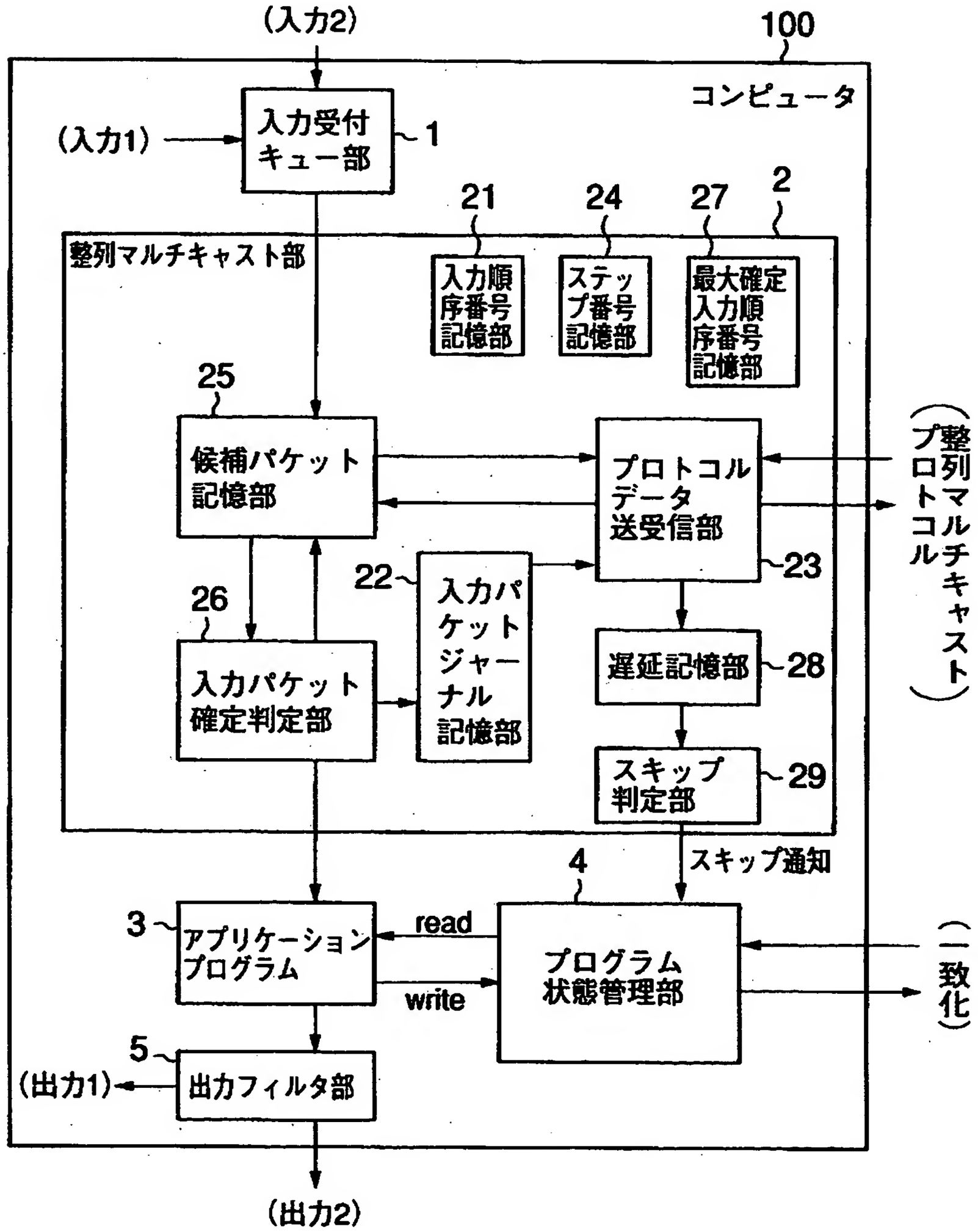
Ｂ…内部ネットワーク

【書類名】　　図面

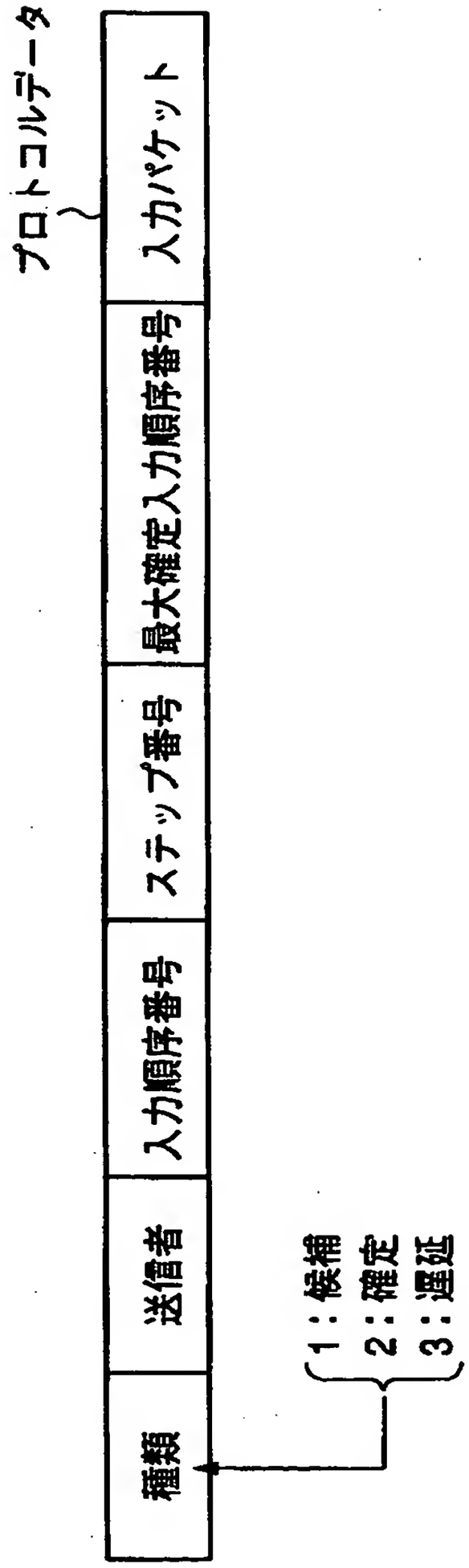
【図１】


A 外部ネットワーク
1000
分散システム
100
コンピュータ
100
コンピュータ
100
コンピュータ
B 内部ネットワーク
2000
クライアント装置
(出力1)
(入力1)
100
コンピュータ
(出力2)
(入力2)

【図２】


(入力2)
100
コンピュータ
(入力1)
入力受付キュー部
1
21
24
27
2
整列マルチキャスト部
入力順序番号記憶部
ステップ番号記憶部
最大確定入力順序番号記憶部
25
候補パケット記憶部
プロトコルデータ送受信部
(整列マルチキャストプロトコル)
22
入力パケットジャーナル記憶部
23
26
入力パケット確定判定部
遅延記憶部
28
スキップ判定部
29
4
スキップ通知
3
アプリケーションプログラム
read
write
プログラム状態管理部
(一致化)
5
(出力1)
出力フィルタ部
(出力2)

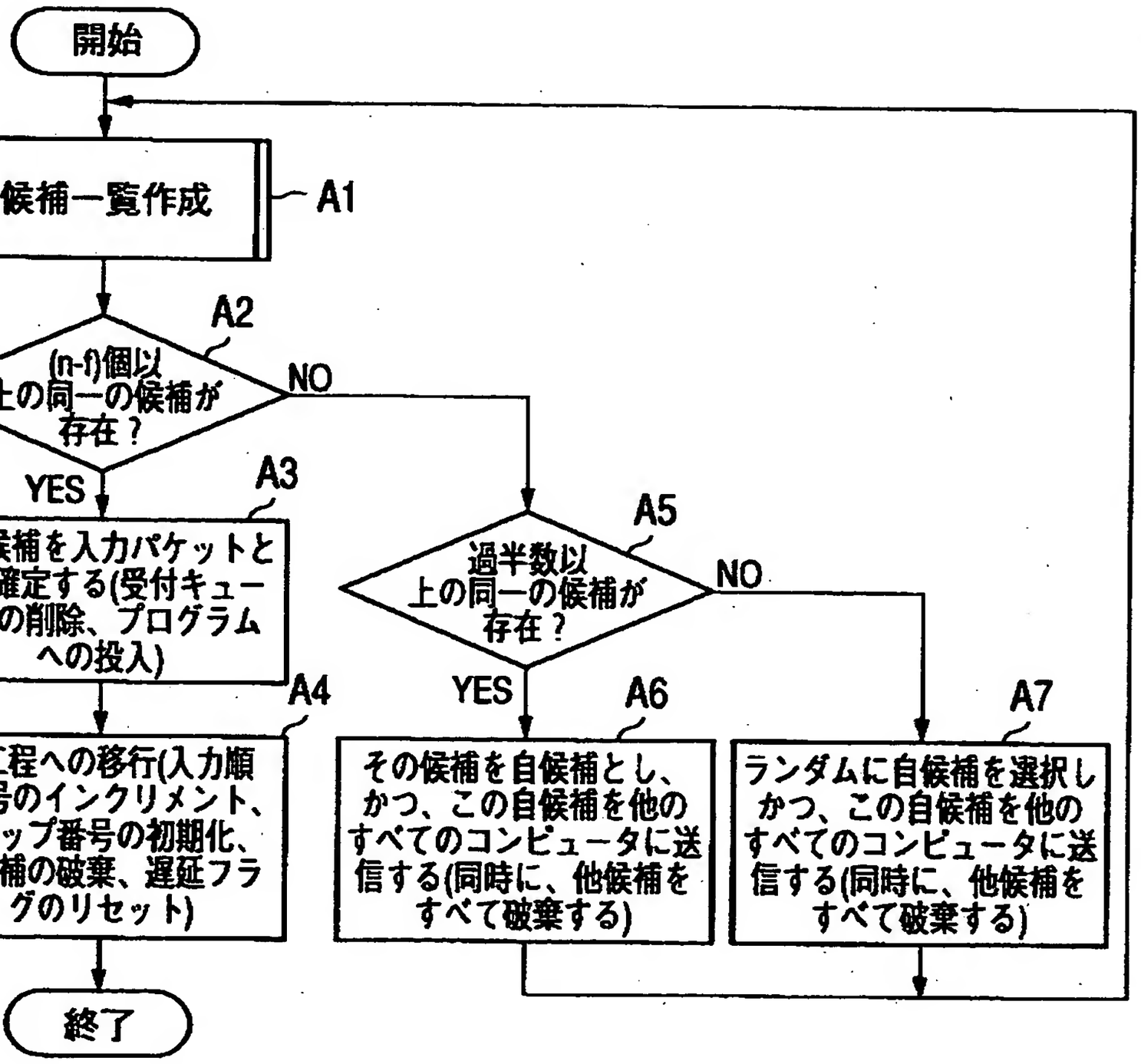
【図３】


プロトコルデータ
入力パケット
最大確定入力順序番号
ステップ番号
入力順序番号
送信者
種類
1：候補
2：確定
3：遅延

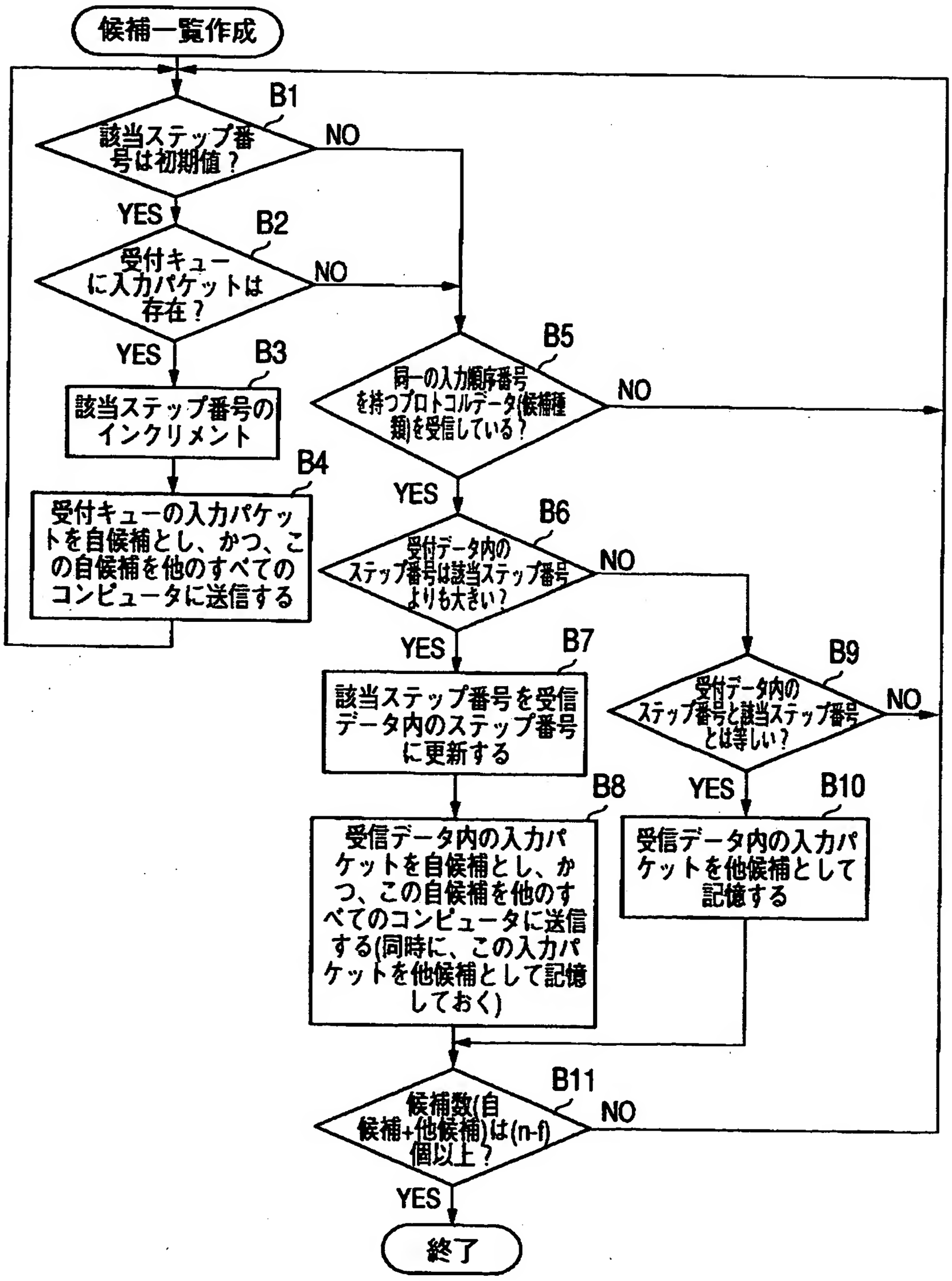
【図４】

＼	コンピュータ(1)	コンピュータ(2)	コンピュータ(3)	コンピュータ(4)
1	A (候補選択)	A (候補選択)	B (候補選択)	C (候補選択)
2	[A(1),A(2),B(3)] (候補一覧作成)	[A(1),A(2),C(4)] (候補一覧作成)	[A(2),B(3),C(4)] (候補一覧作成)	[A(1),A(2),C(4)] (候補一覧作成)
3	A (候補選択)	A (候補選択)	C (候補選択)	A (候補選択)
4	[A(1),A(2),A(4)] (候補一覧作成)	[A(1),A(2),C(3)] (候補一覧作成)	[A(1),A(2),C(3)] (候補一覧作成)	[A(2),C(3),A(4)] (候補一覧作成)
5	決定(A)	A (候補選択)	A (候補選択)	A (候補選択)
6	―――	[A(1),A(2),A(3)] (候補一覧作成)	[A(2),A(3),A(4)] (候補一覧作成)	[A(2),A(3),A(4)] (候補一覧作成)
7	―――	決定(A)	決定(A)	決定(A)
⋮	⋮	⋮	⋮	⋮

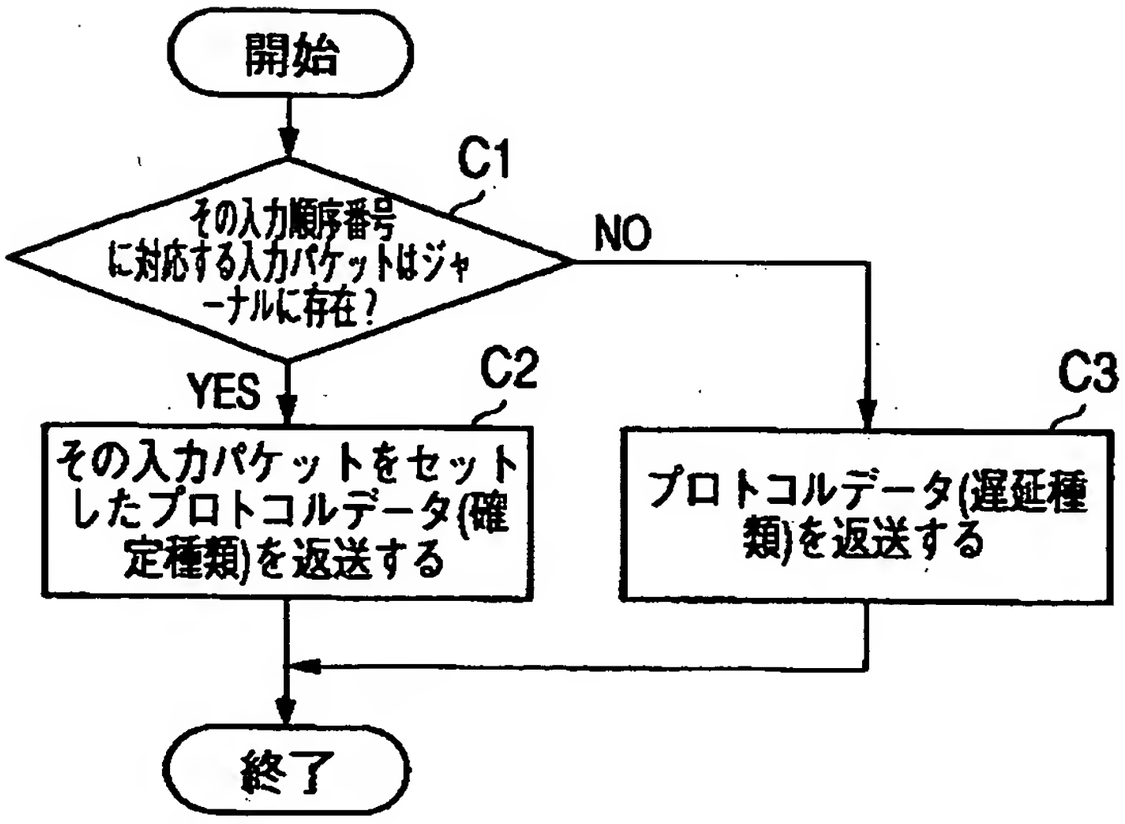
【図５】


開始
候補一覧作成
A1
A2
(n-1)個以上の同一の候補が存在？
NO
YES
A3
その候補を入力パケットとして確定する(受付キューからの削除、プログラムへの投入)
A4
次工程への移行(入力順序番号のインクリメント、ステップ番号の初期化、全候補の破棄、遅延フラグのリセット)
終了
A5
過半数以上の同一の候補が存在？
NO
YES
A6
その候補を自候補とし、かつ、この自候補を他のすべてのコンピュータに送信する(同時に、他候補をすべて破棄する)
A7
ランダムに自候補を選択しかつ、この自候補を他のすべてのコンピュータに送信する(同時に、他候補をすべて破棄する)

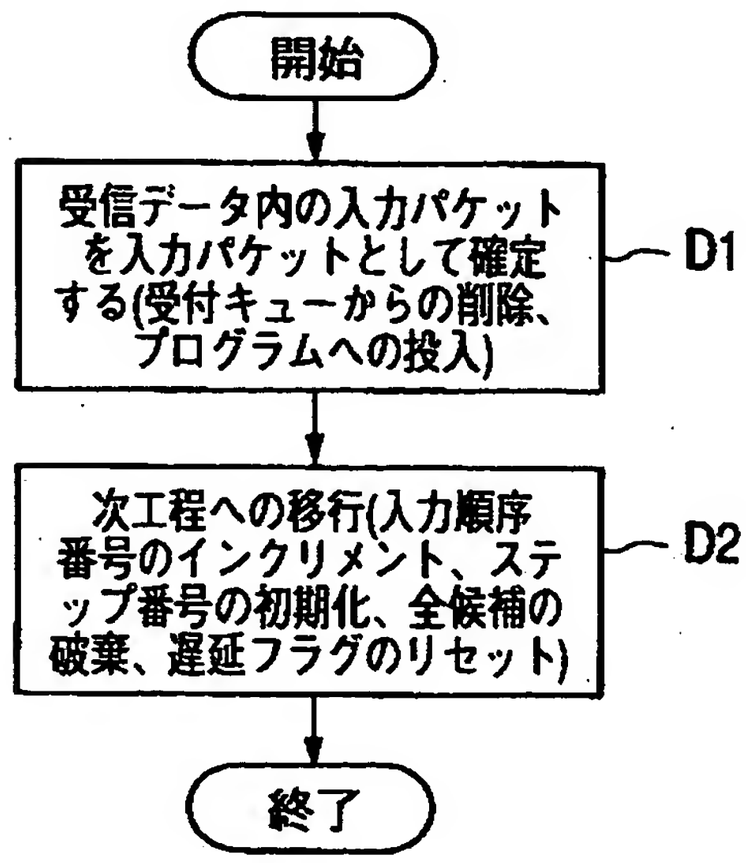
【図６】


候補一覧作成
B1
該当ステップ番号は初期値？
NO
YES
B2
受付キューに入力パケットは存在？
NO
YES
B3
該当ステップ番号のインクリメント
B4
受付キューの入力パケットを自候補とし、かつ、この自候補を他のすべてのコンピュータに送信する
B5
同一の入力順序番号を持つプロトコルデータ(候補種類)を受信している？
NO
YES
B6
受付データ内のステップ番号は該当ステップ番号よりも大きい？
NO
YES
B7
該当ステップ番号を受信データ内のステップ番号に更新する
B8
受信データ内の入力パケットを自候補とし、かつ、この自候補を他のすべてのコンピュータに送信する(同時に、この入力パケットを他候補として記憶しておく)
B9
受付データ内のステップ番号と該当ステップ番号とは等しい？
NO
YES
B10
受信データ内の入力パケットを他候補として記憶する
B11
候補数(自候補+他候補)は(n-f)個以上？
NO
YES
終了

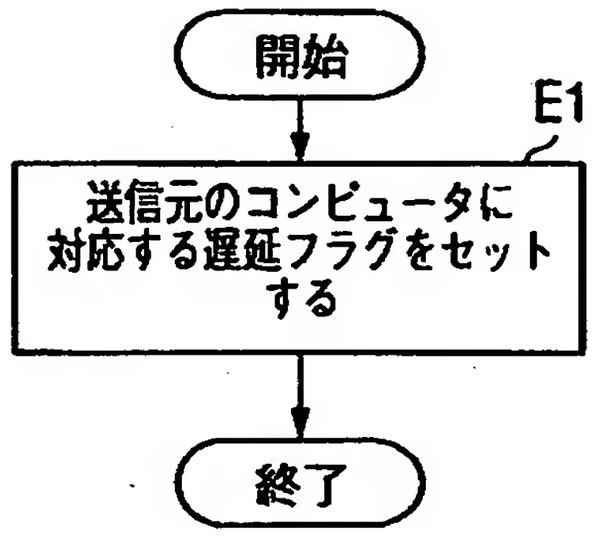
【図７】


開始
C1
その入力順序番号に対応する入力パケットはジャーナルに存在?
NO
YES
C2
その入力パケットをセットしたプロトコルデータ(確定種類)を返送する
C3
プロトコルデータ(遅延種類)を返送する
終了

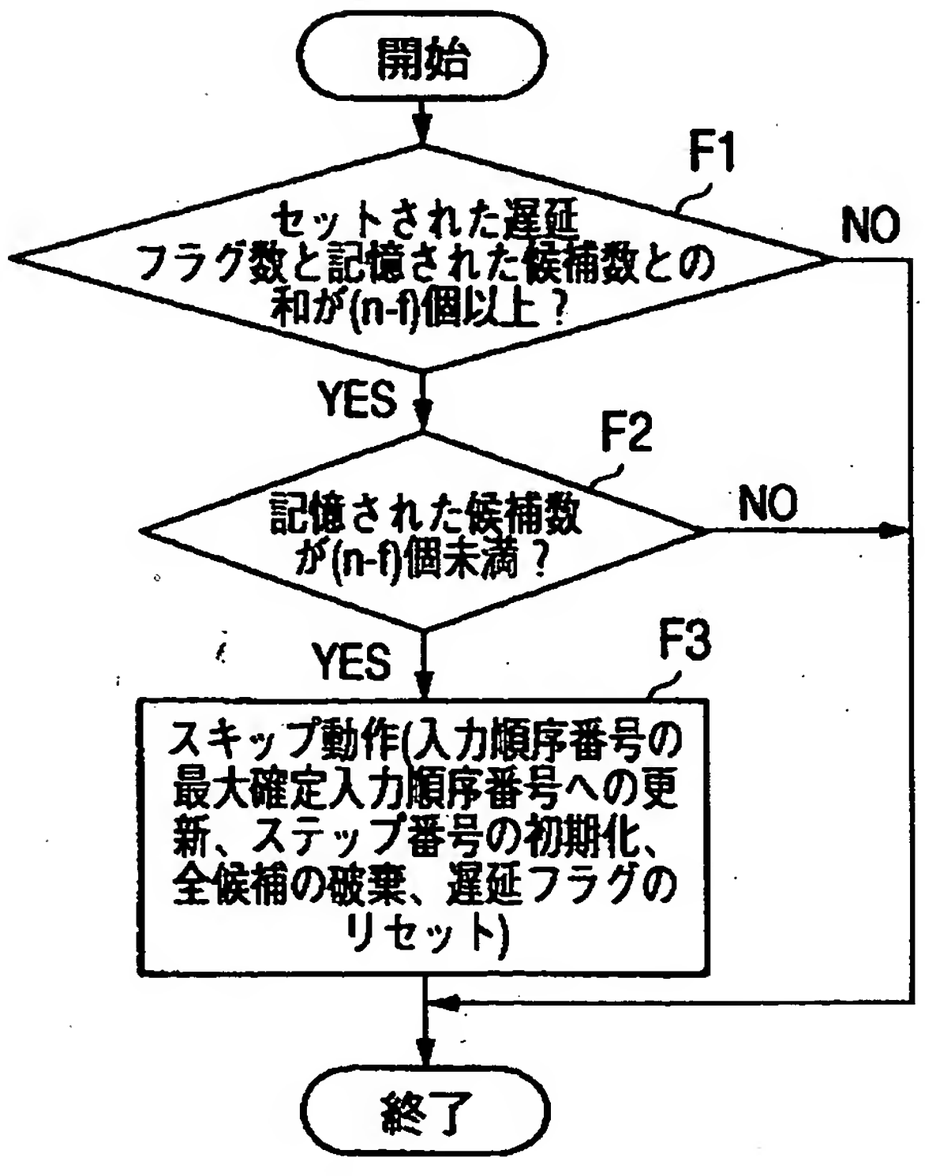
【図８】


開始
受信データ内の入力パケットを入力パケットとして確定する(受付キューからの削除、プログラムへの投入)
D1
次工程への移行(入力順序番号のインクリメント、ステップ番号の初期化、全候補の破棄、遅延フラグのリセット)
D2
終了

【図９】


開始
E1
送信元のコンピュータに対応する遅延フラグをセットする
終了

【図１０】


開始
F1
セットされた遅延フラグ数と記憶された候補数との和が(n-f)個以上？
NO
YES
F2
記憶された候補数が(n-f)個未満？
NO
YES
F3
スキップ動作(入力順序番号の最大確定入力順序番号への更新、ステップ番号の初期化、全候補の破棄、遅延フラグのリセット)
終了

【書類名】　　要約書

【要約】

【課題】スプリットブレインの防止と故障発生時におけるリアルタイム性の確保とを両立させることを可能とした分散システムを提供する。

【解決手段】この分散システムは、ｎ台のコンピュータで多重化を構成し、ｆ台までの故障停止を許容する。そして、各コンピュータは、内部ネットワークＢを介して入力候補を送受信し合い、その一覧を作成する。そして、その中に（ｎ－ｆ）個の同一の入力候補が現れるまで、それぞれがこの一覧作成を繰り返し、この条件を満たしたものから、他のコンピュータの状態に関わらずに、その処理を実行する。つまり、この分散システムは、故障検出をまったく行わないことによって、スプリットブレインを原理的に発生させず、タイムアウトによる故障発生時の処理の中断も発生させることがない。

【選択図】　　図１

出　願　人　履　歴　情　報

識別番号　　〔０００００３０７８〕

１．変更年月日　　１９９０年　８月２２日
　［変更理由］　　新規登録
　　住　所　　神奈川県川崎市幸区堀川町７２番地
　　氏　名　　株式会社東芝

２．変更年月日　　２００１年　７月　２日
　［変更理由］　　住所変更
　　住　所　　東京都港区芝浦一丁目１番１号
　　氏　名　　株式会社東芝